CLAIMS

1. A process for executing programs on a multiprocessor system having a plurality of processors, having a given instruction set architecture, each of said processors being able to execute, at each processing cycle, a respective maximum number of instructions, characterized in that it comprises the operations of:

compiling, at least in part, the instructions of said programs as instruction words of given length executable on a first processor of said plurality; and

modifying at least some of said instruction words of given length converting them into modified-instruction words executable on a second processor of said plurality, said modification operation in turn having at least one operation selected from the group of:

splitting said instruction words into modified-instruction words; and

entering in the modified-instruction words no-operation instructions.

2. The process according to Claim 1, characterized in that it comprises the operations of:

compiling the instructions of said programs in part as first instruction words having a first given length and executable on said first processor of said plurality and in part as second instruction words of given length executable on a second processor of said plurality;

modifying at least some of said first instruction words into first modified-instruction words executable on said second processor of said plurality; and

modifying at least some of said second instruction words into second modified-instruction words executable on said first processor of said plurality.

3. The process according to Claim 2, characterized in that said first instruction words and said second instruction words have, respectively, a first and a second maximum length with said first maximum length greater than said second

maximum length, the quotient between said first maximum length and said second maximum length having a given value with the possible presence of a remainder and in that the procedure comprises the operations of:

modifying said first instruction words by having said first maximum length into first modified-instruction words having said second maximum length by:

splitting said first instruction words into a number of said first modified-instruction words equal to the value of said quotient; and

in the presence of said remainder, adding to said first modified-instruction words a further modified-instruction word of length equal to said second maximum length, said second maximum length being obtained by entering into said further first modified-instruction word a set of no-operation instructions; and

modifying said second instruction words by having said second maximum length into second modified-instruction words having said first maximum length by:

adding to said second instruction words of said second maximum length a number of no-operation instructions equal to the difference between said first maximum length and said second maximum length.

4. The process according to Claim 1 characterized in that it comprises the operations of:

encoding said instructions on a given number of bits, said number of bits having a first bit identifying a length of instruction word executable on a processor of said plurality;

associating to said given number of bits a respective appendix having a set of further bits identifying lengths of instruction words executable on different processors of said plurality;

identifying for each of said instructions a processor of said plurality designed to execute said instruction, said identified processor being able to process for each processing cycle a given length of instruction word; and

entering in the position of said first identifier bit a chosen bit between said further bits of said appendix, said chosen bit identifying the length of instruction word that can be executed by said identified processor.

5. The process according to Claim 4, characterized in that it comprises the operation of erasing said respective appendix before execution of the instruction.

6. The process according to Claim 4, characterized in that said chosen bit is entered in the position of said first identifier bit in a step chosen from among:

decoding of the instruction in view of the execution;

re-filling of the cache associated to said identified processor; and

decompression of the instruction in view of the execution.

7. The process according to Claim 1, characterized in that it comprises the operation of:

alternatively distributing the execution of the instructions of said sequence between the processors of said plurality, said instructions being directly executable by the processors of said plurality in conditions of binary compatibility.

8. The process according to Claim 1, characterized in that it comprises the operation of selectively distributing the execution of said instructions among the processors of said plurality, distributing dynamically the computational load of said processors.

9. The process according to Claim 1, characterized in that it comprises the operation of selectively distributing the execution of said instructions

between said processors of said plurality with the criterion of equalizing the operating frequency of the processors of said plurality.

10. The process according to Claim 1, characterized in that it comprises the operation of performing a control process executed by at least one of the processors of said plurality so as to equalize its own workload with respect to the other processors of said multiprocessor system.

11. The process according to Claim 1, characterized in that it comprises the operation of drawing up a table accessible by said control process, said table having items selected from the group of:

a list of processes being executed or suspended on any processor of said plurality of processors;

the progressive number thereof according to the order of activation;

the percentage of maximum power of the processor that is used by said process;

the execution time, said time, if zero, indicating that the process is temporarily suspended from being executed;

the amount of memory of the system used by the process to be able to execute the function for which it is responsible;

the maximum length of the long instruction that the VLIW processor can execute and for which it had been generated during compiling;

maximum length of the long instruction of the VLIW processor on which it is executed; and

the address of the portion of memory in which the data and the instructions are stored.

12. A multiprocessor system, configured for operating with the process according to Claim 1.

13. The multiprocessor system according to Claim 12, characterized in that said processors are all of the VLIW type.

14. The multiprocessor system according to Claim 12, characterized in that said plurality of processors comprises at least one VLIW processor and at least one superscalar processor.

15. A system comprising:

a plurality of processors coupled for receiving instruction sets;

a first processor of the plurality coupled to each of the other processors within said plurality, said first processor receiving from the other processors data representative of the workload of each of said other processors;

an output signal from said first processor to said instruction set stream, said output signal controlling the instructions, which are sent to each of said processors based on the results of the workload measurement of said processors.

16. The system according to Claim 15, wherein said workload measurement comprises power consumption of each of said processors of said plurality.

17. The system according to Claim 15, wherein said workload measurement comprises memory usage of each of said processors of said plurality.

18. The system according to Claim 15, wherein said workload measurement comprises number of operations carried out by each of said processors of said plurality.

19. A process of directing instruction sets to be executed by a plurality of processors in a system comprising:

receiving a plurality of instruction sets on a bus line connected to said processors;

receiving workload data at a first processor of said plurality of processors, said workload data being representative of workload of each of the processors of said plurality;

comparing the workload of each of the processors; and

sending a signal from said first processor based on the data representative of the workload of each of the processors of said plurality to the bus line for modifying the number of instruction sets sent to each processor based on their respective workloads.

20. The process according to Claim 19, wherein said workload data includes data regarding power consumption of each of said processors of said plurality.

21. The system according to Claim 19, wherein said workload data includes data regarding memory usage of each of said processors of said plurality.

22. The system according to Claim 19, wherein said workload data includes data regarding the number of operations carried out by each of said processors of said plurality.

23. A process for executing programs in a system having a plurality of processors comprising:

receiving instructions of said programs, a first set of said instructions having a first word length, and second set of said instructions having a second word length, said first word length being longer than said second word length;

modifying the length of said second instruction words by combining two or more of said second instruction words into one instruction word, the words selected for combining being selected to ensure that the length of the combined instruction word does not exceed said first word length;

adding no-operation instructions equal to the difference between said first word length and the combined instruction word length to said combined instruction word.

24. The process according to Claim 23, wherein said first word length is equal to an exact multiple of said second word length, such that the combined instruction word equals the first word length and no no-operation instructions are added.

25. The process according to Claim 23, wherein said instructions are further modified comprising:

encoding said instructions on a given number of bits, said number of bits having a first bit identifying the number of no-operation instructions within the instruction word.